

Standard disseminators

Status

This page is still **volatile**:

- While we have a good general idea about the default disseminator, the specifics of its behaviors are still under investigation.
- The metadata disseminator is stable.
- The master file disseminator has not been implemented.

All objects will subscribe to the default disseminator and the metadata disseminator. Most objects will also subscribe to one or more type-specific disseminators.

Default disseminator
getFullAssetDefinition
getLabel
getPreview
getFullView
getDefaultView
Metadata disseminator
getMetadata(format)
Master file disseminator
getMasterFile

Default disseminator, for all objects (bdef:iudlDefault)

The default disseminator is heavily based on the current "loose standard" for [asset actions](#) as well as the descriptions in [Default Disseminator Design](#). This disseminator will be modified as we gain more experience with our needs for cross-collection functionality.

getFullAssetDefinition – Returns the current version of the asset definition, in XML form.

getLabel – Returns the appropriate string of plain text (UTF-8) that can be included in a list of labels (title)

getPreview – Returns a media file that provides a simple preview of the resource. This media file can be any MIME type. The specific MIME type of the object can be determined by looking at the asset definition. The preview is a view of the resource that can share the viewing space with other previews, suitable for viewing in the context of any collection

getFullView – Returns the appropriate HTML view of the resource when it has the whole viewing space to itself, rendered in terms of its default collection. This is the same representation that is given by the item-level PURL for the object.

getDefaultView – Returns a byte stream of raw content for the resource, the most common view. In many cases, this may be the same as getFullView, but like the preview it can be any MIME type.

Metadata disseminator, for all objects (bdef:iudlMetadata)

This disseminator is separate from the default disseminator because it may need more customization from collection to collection.

getMetadata(format) – Returns a metadata record describing this object. The default is the full METS-formatted metadata stream stored in this object. If a format is specified, a portion of the METS document is returned. The returned portion is a single section of the METS document, retrieved by its ID.

Acceptable formats include:

- mets – return the full METS document
- dmdSec-mods – return a MODS description of this item (if available)
- dmdSec-dc – return a DC description of this item
- other identifiers from the METS document

Implementation

Prerequisites:

- Any Fedora that uses this disseminator must have a datastream containing a METS document.
- The util:METS-XSL object must be in the repository.

Files:

- [bdef:iudlMetadata.xml](#) - behavior definition

- [bmech:iudlMETSMetadata.xml](#) - behavior mechanism
- [util:METS-XSL.xml](#) - utility object containing XSL needed by the disseminator

Master file disseminator, for all objects (bdef:iudlMasterFile)

Even though this disseminator will always be used along with other basic file disseminators (like the image disseminator or audio disseminator), we have separated this disseminator from others for several reasons:

- Master files often have different access control needs than derivative files.
- Master files are often used differently than derivative files. They are not accessed on a regular basis, although they must be accessed by preservation-integrity tools.
- Our implementation will be very specific to the computing environment at Indiana University. Other institutions will likely not want to use it.

This disseminator should always be attached to objects at the "leaf level". That is, the objects that actually hold media files, not objects that aggregate other objects. It is assumed that there is only one master file for each object this disseminator is attached to.

getMasterFile – Requests delivery of the master media file for this object. This may take some time if the file is in HPSS, so the resultant file will be emailed to the user or placed in a temporary location. This method can only be run by privileged users.